

Esame di Calcolatori Elettronici T

28 Gennaio 2021 (Ing. Informatica)

Esercizio 1

Progettare un sistema, basato su un processore DLX dotato di 512 MB di EPROM mappata agli indirizzi bassi e 2 GB di RAM mappata agli indirizzi alti. Avendo già progettato una porta in input a 8 bit, in grado di comunicare attraverso il protocollo di *handshake*, si desidera, mediante la strategia delineata in seguito, leggere da tre dispositivi esterni di taglia diversa che comunicano con handshake: uno a 8 bit (mediante INPUT_PORT_8), uno a 16 bit (mediante INPUT_PORT_16, da progettare) e uno a 32 bit (medinate INPUT_PORT_32, da progettare).

Allorché tutte le porte in input siano pronte per eseguire un trasferimento, i dati letti dalle stesse, estesi a 32 bit con segno, dovranno essere memorizzati come *word* agli indirizzi seguenti: FFFF0100h (INPUT_PORT_8), FFFF0200h (INPUT_PORT_16) e FFFF0300h (INPUT_PORT_32). I trasferimenti, disabilitati all'avvio del sistema, devono poter essere abilitati/disabilitati mediante appropriati comandi software.

- Descrivere sinteticamente la soluzione indicando chiaramente quali sono i dispositivi utilizzati, gli indirizzi e i segnali di *chip-select*
- Usando la porta a 8 bit, progettare INPUT_PORT_16 e INPUT_PORT_32
- Progettare il sistema in base alle specifiche del testo minimizzando le risorse necessarie ed evidenziando/gestendo eventuali criticità
- Indicare le espressioni di decodifica e il range di indirizzi di tutte le periferiche, le memorie e i segnali e le connessioni di tutti i dispositivi con i bus di sistema
- Scrivere il codice che consente di eseguire i trasferimenti indicati in precedenza. Si assuma che per questa finalità i registri da R20 a R25 possano essere utilizzati senza la necessità di doverli ripristinare
- Scrivere il codice per attivare i trasferimenti dalle porte in input

Esercizio 2

Il codice seguente è corretto? Motivare chiaramente la risposta.

```
ADDI R7,R0,4321h
SUBI R8,R0,0001h
X LHU R9,R7(1022h) ← Si legge hw a un indirizzo
AND R10,R9,R8      ← dispari
```

Esercizio 3

In una CPU *pipelined*: come sono eliminate le istruzioni erroneamente inserite nella pipeline di elaborazione nel caso la predizione di un salto risulti errata?

- Vengono sostituite con delle NOP

Risposte vaghe e/o non focalizzate sulle domande del testo non saranno MINIMAMENTE considerate.

MAPPING E DECODIFICA

512 KB EPROM		2GB RAM
--------------	--	---------

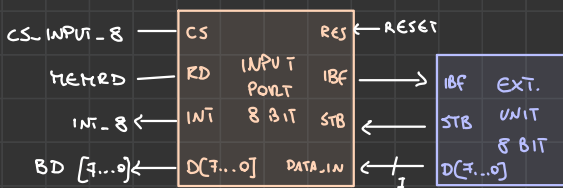
CS_RAM_2GB_0 = BA31 BE0
 " - 1 = " BE1
 " - 2 = " BE2
 " - 3 = " BE3

CS_EPROM_512KB_0 = BA31 BA30 BE0
 " - 1 = " BE1
 " - 2 = " BE2
 " - 3 = " BE3

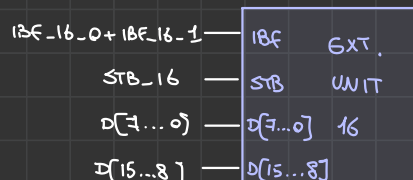
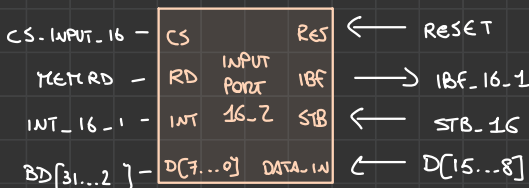
CS_INPUT_8 = BA31 BA30 BA2 BE0
 CS_INPUT_16 = " " BE2 BE3
 CS_INPUT_32 = " BA2
 CS_ENABLE_INT = " BA2 BE1

0x40000000
 0x40000003
 0x40000004
 0x40000001

PORTA DI INPUT

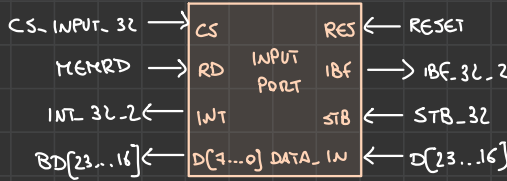
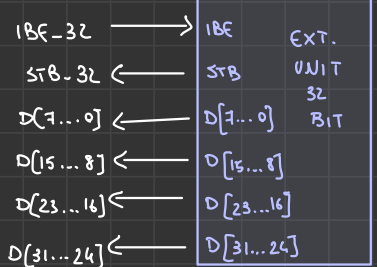
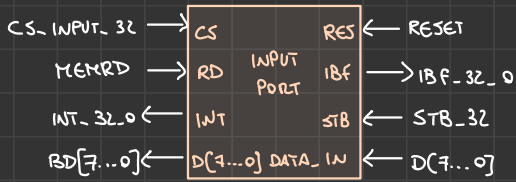


PORTA 16 BIT

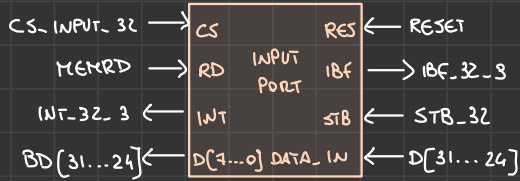


INT_16 = INT_16_0 + INT_16_1

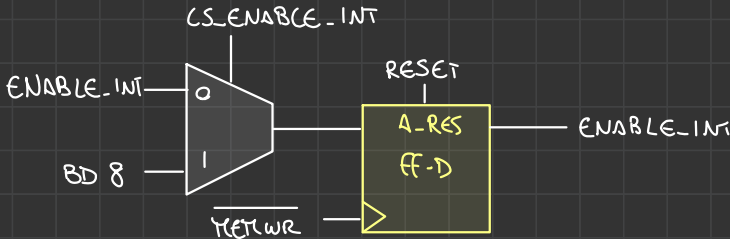
PORTA A 32 BIT



$$IBF_{32} = IBF_{32_0} + IBF_{32_1} + IBF_{32_2} + IBF_{32_3}$$



$$INT_{32} = INT_{32_0} \cdot INT_{32_1} \cdot INT_{32_2} \cdot INT_{32_3}$$



$$INT(TO\ DLX) = ENABLE_INT \cdot INT_8 \cdot INT_16 \cdot INT_32$$

CODICE DEI TRASFERIMENTI

CS-INPUT_8	0x4000000	
CS-INPUT_16	0x4000002/3	INPUT_8 → FFFF0100 h
CS-INPUT_32	0x4000004	INPUT_16 → FFFF0200 h
CS-ENABLE_INT	0x4000001	INPUT_32 → FFFF0300 h

0h LHI R20, 0x4000 ; Prepara indirizzo per leggere dalla input-port
4h LHI R21, 0xFFFF ; Indirizzo per scrivere i dati
8h LB R22, 0x0000(R20); Leggo da PORT_8 un byte
Ch SW R22, 0x0100(R21); Memorizzo su 0xFFFF0100 un word
10h LH R22, 0x0002(R20); leggo da PORT_16 un half-word
14h SW R22, 0x0200(R21); Memorizzo su 0xFFFF0200 un word
18h LW R22, 0x0004(R20); Leggo da PORT_32 un word
1Ch SW R22, 0x0300(R21); Memorizzo su 0xFFFF0300 un word

CODICE PER ATTIVARE I TRASFERIMENTI

1000h LHI R7, 0x4000 ; Prepara indirizzo CS-ENABLE_INT
1004h ADDUI R8, R0, 0x0001; R8 = 0x00000001
1008h SB R8, 0x0001(R8); Pone ENABLE_INT a 1

MEMORIE

$$\text{RAM } 2\text{GB} = 2 \cdot 1024^3 \text{ Byte} = 2^{31} \text{ Byte}$$

